

第一次数据库系统作业

学号：1711350

班级：计算机科学与技术一班

姓名：李伟

第一题：数据库设计初步：

一、连锁便利店数据库需求：

1 日常经营行为需求：

- 1.1 商品能够录入和退出系统：为了满足超市日常经营的基本要求，系统要实现新商品信息的录入操作，对于库存中已经存在的商品，能够在进货时实现一键扫描加数量的简单方式实现增加库存的操作，同时对于过期或者停止销售的商品，能够实现从系统中删除或者标记其特殊状态的功能。
- 1.2 零售业务：收银员能够通过扫描条形码以及输入商品名称等方式查询添加顾客购买的商品信息，生成购物清单，并计算商品总价，在获得顾客的现金后能够自动生成找零以及商品清单信息，对于销售出去的商品能够在数据库中记录商品的销售信息。对于持有便利店VIP卡的顾客还可以自动计算折扣，并扣除会员账户余额，记录会员消费金额，计算积分，对于有时间期限的会员能够自动识别会员是否到期。
- 1.3 退货信息管理：遇到客户进行退货时，能够根据购物清单及时查询相关交易信息并进行退货处理，记录退货时间和退货金额（全额退款或部分退款等情况）等信息。

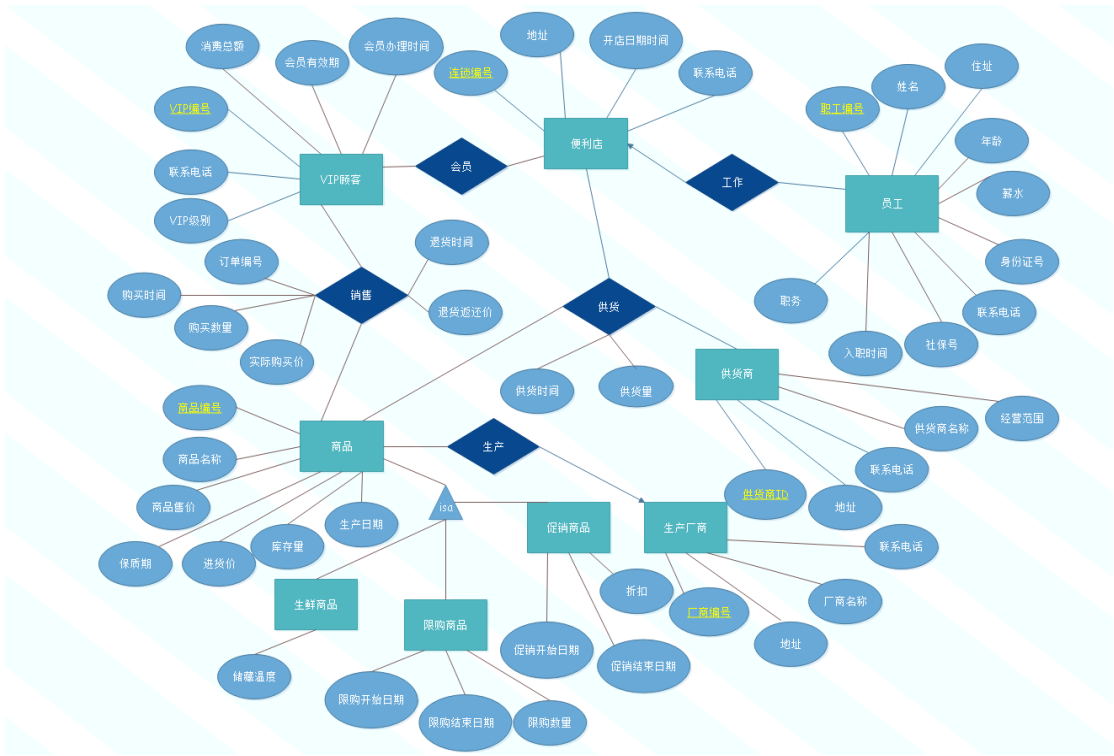
2 管理行为需求

- 2.1 销售管理：商品的正常销售，促销和限购。综合查询各种销售记录，以及账面情况，能够利用数据库生成各种销售报表。
- 2.2 库存管理：可以利用数据库查询商品的库存和销售信息，并生成各种商品的销量情况，从而能够据此制定销售计划。另外可以根据数据库信息对商品过期以及库存不足情况进行监控，自动进行预警。进一步可以实现库存的自动盘点。
- 2.3 进货管理：能够记录和生成各批次进货的明细情况，包括进货时间、进货商品品类和进货价格、供货商信息等，在货物出现问题后能够及时查询进货信息追更溯源。可以根据销售情况和库存信息，人工或自动制定进货计划。
- 2.4 人事信息管理：能够对便利店的员工各种信息进行综合管理，能够对不同的工作职位设置不同的数据库系统权限，可以综合查询员工、会员、供货商、商品厂商等的基本信息。

二、采用教材中的方法实现以上数据库的设计：

- 1 由上述需求可画出连锁便利店的数据库概念模型ER图如图1所示(画图采用EDrawn软件实现，主键采用黄色字体和下划线标记)；
- 2 将上述ER图转换为关系模式如下（主键用下划线和加粗字体标识，外键采用下划线斜体字体标识，参照在备注中说明，子类表的设计采用ER的方式）：
 - 2.1 便利店信息表（**连锁编号**，地址、开店日期时间、联系电话）；
 - 2.2 员工信息表（**职工编号**，*工作便利店编号*，姓名，住址，年龄，薪水，身份证号，联系电话，社保号，入职时间，职位）；将工作的联系合并到员工信息表，“工作便利店编号”参照便利店信息表中的连锁编号。
 - 2.3 供货商信息表（**供货商ID**，地址，联系电话，供货商名称，经营范围）；
 - 2.4 VIP顾客信息表（**VIP编号**，联系电话，VIP级别，消费总额，会员有效期，会员办理时间）；
 - 2.5 生产厂商信息表（**厂商编号**，地址，联系电话，厂商名称）；

- 2.6 商品信息表（商品编号，生产厂商编号，商品名称，商品售价，商品进货价，生产日期，保质期，库存量）；合并生产联系，生产厂商编号需参照生产厂商信息表中的厂商编号；
- 2.7 生鲜商品信息表（商品编号，储藏温度）；这里的子类商品编号需参照商品信息表中的商品编号，下面两种商品信息表与之相同。
- 2.8 限购商品信息表（商品编号，限购数量，限购开始日期，限购结束日期）；
- 2.9 促销商品信息表（商品编号，折扣，促销开始日期，促销结束日期）；
- 2.10 供货联系表（便利店连锁编号，供货商 ID，商品编号，供货时间，供货量）；这里的三个编号分别需要参照便利店、供货商、商品信息表中的编号；
- 2.11 会员联系表（连锁便利店编号，VIP 编号）；连锁便利店编号和VIP 编号需要分别参照便利店信息表的便利店连锁编号和VIP 顾客信息表中的VIP 编号；
- 2.12 销售联系表（VIP 编号，商品编号，订单编号，购买时间，购买数量，实际购买价，退货时间，退货返还价）；VIP 编号参照VIP 顾客信息表的VIP 编号，商品编号参照商品信息表的商品编号。



图表 1 连锁便利店 ER 图示

3 数据库表结构设计和说明：

3.1 数据库表属性设计：

图表 2 数据库表检索

表名	中文名
Store	便利店信息表
Employee	员工信息表
VIP	VIP 顾客信息表
Supplier	供货商信息表
Manufacturer	生产厂商信息表

Commodity	商品信息表
FreshCommodity	生鲜商品信息表
LimitCommodity	限购商品信息表
DiscountCommodity	促销商品信息表
Sale	销售联系表
Supply	供货联系表
Register	会员联系表

图表 3 便利店（Store）信息表

字段名	字段类型	长度	主\外键	字段约束	对应中文名
Store_ID	Int	4	Primary	Not null	连锁编号
Store_address	Varchar	256		Not null	地址
Store_openDate	Datetime	8		Not null	开店日期
Store_teleNO	varchar	25		Not null	联系电话

图表 4 员工（Employee）信息表

字段名	字段类型	长度	主\外键	字段约束	对应中文名
Employee_ID	Int	4	Primary	Not null	职工编号
Store_ID	Int	4	Foreign	Not null	便利店编号
Employee_name	varchar	20		Not null	姓名
Employee_address	varchar	256		Null	住址
Employee_age	Int	4		Null	年龄
Employee_salary	Int	4		Not null	薪水
Employee_IDNO	varchar	25		Not null	身份证号
Employee_teleNO	varchar	25		Not null	联系电话
Employee_SocialSecurity	varchar	25		Not null	社保号
Employee_enterDate	datetime	8		Not null	入职时间
Employee_position	varchar	40		Not null	职位

图表 5 供应商（Supplier）信息表

字段名	字段类型	长度	主\外键	字段约束	对应中文名
Supplier_ID	int	4	Primary	Not null	供货商 ID
Supplier_address	varchar	256		Not null	地址
Supplier_teleNO	varchar	25		Not null	联系电话
Supplier_name	varchar	20		Not null	供货商名称
Supplier_business	Varchar	1024		Null	经营范围

图表 6 会员（VIP）信息表

字段名	字段类型	长度	主\外键	字段约束	对应中文名
VIP_ID	int	4	Primary	Not null	VIP 编号
VIP_teleNO	varchar	25		Not null	联系电话
VIP_level	Int	4		Not null	VIP 级别

VIP_consumption	int	4		Null	消费总额
VIP_validity	datetime	8		Null	会员有效期
VIP_datetime	datetime	8		Not null	会员办理时间

图表 7 生产厂商（manufacture）信息表

字段名	字段类型	长度	主\外键	字段约束	对应中文名
Manf_ID	Int	4	Primary	Not null	厂商编号
Manf_address	Varchar	256		Not null	地址
Manf_teleNO	Varchar	25		Not null	联系电话
Manf_name	varchar	50		Not null	厂商名称

图表 8 商品（commodity）信息表

字段名	字段类型	长度	主\外键	字段约束	对应中文名
Cmd_ID	Int	4	Primary	Not null	商品编号
Manf_ID	Int	4	Foreign	Not null	生产厂商编号
Cmd_name	varchar	25		Not null	商品名称
Cmd_saleprice	Int	4		Not null	商品售价
Cmd_buyprice	Int	4		Not null	商品进货价
Cmd_proDate	datetime	8		Not null	生产日期
Cmd_leftNum	int	4		Not null	库存量
Cmd_shelflife	datetime	8		Not null	保质期

图表 9 生鲜商品（FreshCommodity）信息表

字段名	字段类型	长度	主\外键	字段约束	对应中文名
FCmd_ID	Int	4	P and F	Not null	商品编号
FCmd_temp	Int	4		Not null	储藏温度

图表 10 折扣商品（disCountcommodity）信息表

字段名	字段类型	长度	主\外键	字段约束	对应中文名
DisCmd_ID	Int	4	P and F	Not null	商品编号
DisCmd_disc	Int	4		Not null	折扣
DisCmd_discStart	datetime	8		Not null	折扣开始
DisCmd_discEnd	datetime	8		Not null	折扣结束

图表 11 限购商品（limitCommodity）信息表

字段名	字段类型	长度	主\外键	字段约束	对应中文名
limitCmd_ID	int	4	P and F	Not null	商品编号
limitCmd_num	int	4		Not null	限购数量
limitCmd_start	datetime	8		Not null	限购开始
limitCmd_end	datetime	8		Not null	限购结束

图表 12 供货 (supply) 联系表

字段名	字段类型	长度	主\外键	字段约束	对应中文名
Store_ID	Int	4	P and F	Not null	便利店编号
Supplier_ID	Int	4	P and F	Not null	供货商 ID
Cmd_ID	Int	4	P and F	Not null	商品编号
Supply_time	Datetime	8		Not null	供货时间
Supply_num	int	4		Not null	供货数量

图表 13 会员注册 (register) 联系表

字段名	字段类型	长度	主\外键	字段约束	对应中文名
Store_ID	Int	4	P and F	Not null	便利店编号
VIP_ID	Int	4	P and F	Not null	VIP 编号

图表 14 销售 (sale) 联系表

字段名	字段类型	长度	主\外键	字段约束	对应中文名
VIP_ID	Int	4	P and F	Not null	VIP 编号
Cmd_ID	Int	4	P and F	Not null	商品编号
Sale_orderID	Int	4		Not null	订单编号
Sale_time	Datetime	8		Not null	购买时间
Sale_num	Int	4		Not null	购买数量
Sale_price	Int	4		Not null	实际购买价
Sale_returnTime	Datetime	10		Null	退货时间
Sale_returnPrice	Int	4		Null	退货返还价

3.2 数据库表关系模式创建 (SQL) 【代码文件见附件 “CCStore_SQLQuery1.sql”】:

```
CREATE TABLE Store(
    Store_ID int Not null,
    Store_address varchar(256) Not null,
    Store_openDate datetime Not null,
    Store_teleNO varchar(25) Not null,
    primary key(Store_ID)
);
```

```
CREATE TABLE Employee(
    Employee_ID int Not null,
    Store_ID int Not null,
    Employee_name varchar(20) Not null,
    Employee_address varchar(256) null,
    Employee_age int null ,
    Employee_teleNO varchar(25) Not null,
    Employee_salary int Not null,
    Employee_IDNO varchar(25) Not null,
    Employee_SocialSecurity varchar(25) Not null,
```

```
Employee_enterDate  datetime Not null,  
Employee_position  varchar(40) Not null,  
primary key(Employee_ID),  
foreign key(Store_ID) references Store(Store_ID)  
);
```

```
CREATE TABLE Supplier(  
    Supplier_ID  int Not null,  
    Supplier_name varchar(20) Not null,  
    Supplier_address varchar(256) Not null,  
    Supplier_teleNO varchar(25) Not null,  
    Supplier_business varchar(1024) null,  
    primary key(Supplier_ID)  
);
```

```
CREATE TABLE VIP(  
    VIP_ID  int Not null,  
    VIP_level int Not null,  
    VIP_consumption int null ,  
    VIP_teleNO varchar(25) Not null,  
    VIP_validity datetime null,  
    VIP_datetime datetime Not null,  
    primary key(VIP_ID)  
);
```

```
CREATE TABLE Manufacture(  
    Manf_ID  int Not null,  
    Manf_name varchar(50) Not null,  
    Manf_address varchar(256) Not null,  
    Manf_teleNO varchar(25) Not null,  
    primary key(Manf_ID)  
);
```

```
CREATE TABLE Commodity(  
    Cmd_ID  int Not null,  
    Manf_ID int Not null,  
    Cmd_saleprice int Not null,  
    Cmd_buyprice int Not null,  
    Cmd_name varchar(25) Not null,  
    Cmd_proDate datetime Not null,  
    Cmd_leftNum int,  
    Cmd_shelflife datetime Not null,  
    primary key(Cmd_ID),
```

```

        foreign key (Manf_ID) references Manufacture (Manf_ID)
    );

CREATE TABLE FreshCommodity(
    FCmd_ID int Not null,
    FCmd_temp int Not null,
    primary key (FCmd_ID),
    foreign key (FCmd_ID) references Commodity (Cmd_ID)
);

CREATE TABLE DiscountCommodity(
    DisCmd_ID int Not null,
    DisCmd_disc int Not null,
    DisCmd_discstart datetime Not null,
    DisCmd_discend datetime Not null,
    primary key (DisCmd_ID),
    foreign key (DisCmd_ID) references Commodity (Cmd_ID)
);

CREATE TABLE LimitCommodity(
    LimitCmd_ID int Not null,
    LimitCmd_num int Not null,
    LimitCmd_start datetime Not null,
    LimitCmd_end datetime Not null,
    primary key (LimitCmd_ID),
    foreign key (LimitCmd_ID) references Commodity (Cmd_ID)
);

CREATE TABLE Supply(
    Store_ID int Not null,
    Supplier_ID int Not null,
    Cmd_ID int Not null,
    Supply_time datetime Not null,
    Supply_num int Not null,
    primary key (Store_ID, Supplier_ID, Cmd_ID),
    foreign key (Store_ID) references Store (Store_ID),
    foreign key (Supplier_ID) references Supplier (Supplier_ID),
    foreign key (Cmd_ID) references Commodity (Cmd_ID)
);

CREATE TABLE Register(
    Store_ID int Not null,
    VIP_ID int Not null,

```

```

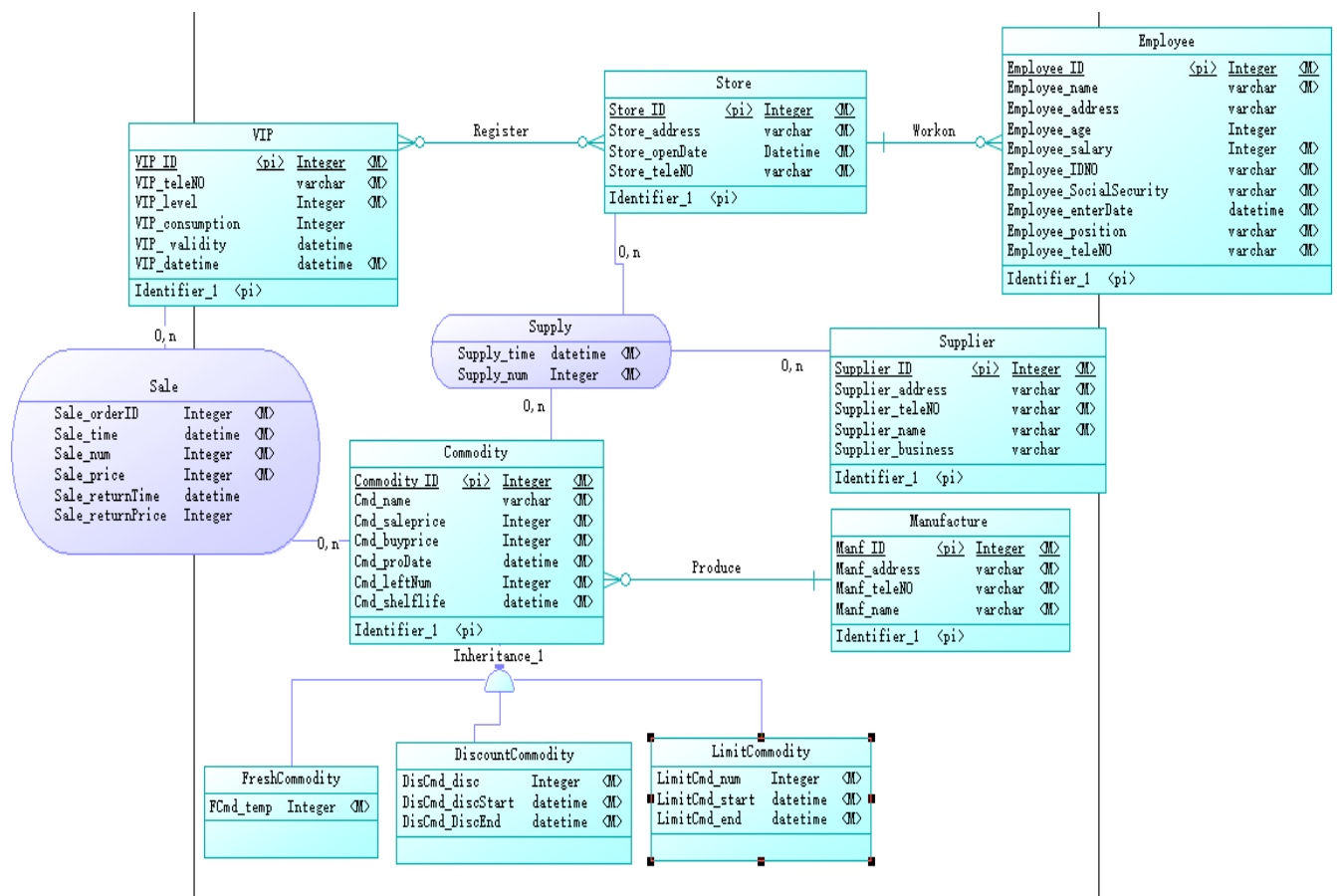
primary key(Store_ID, VIP_ID),
foreign key(Store_ID) references Store(Store_ID),
foreign key(VIP_ID) references VIP(VIP_ID)
);

CREATE TABLE Sale(
VIP_ID int Not null,
Cmd_ID int Not null,
Sale_orderID int Not null,
Sale_time datetime Not null,
Sale_num int Not null,
Sale_price int Not null,
Sale_returnTime datetime null,
sale_returnprice int null ,
primary key(VIP_ID, Cmd_ID),
foreign key(VIP_ID) references VIP(VIP_ID),
foreign key(Cmd_ID) references Commodity(Cmd_ID)
);

```

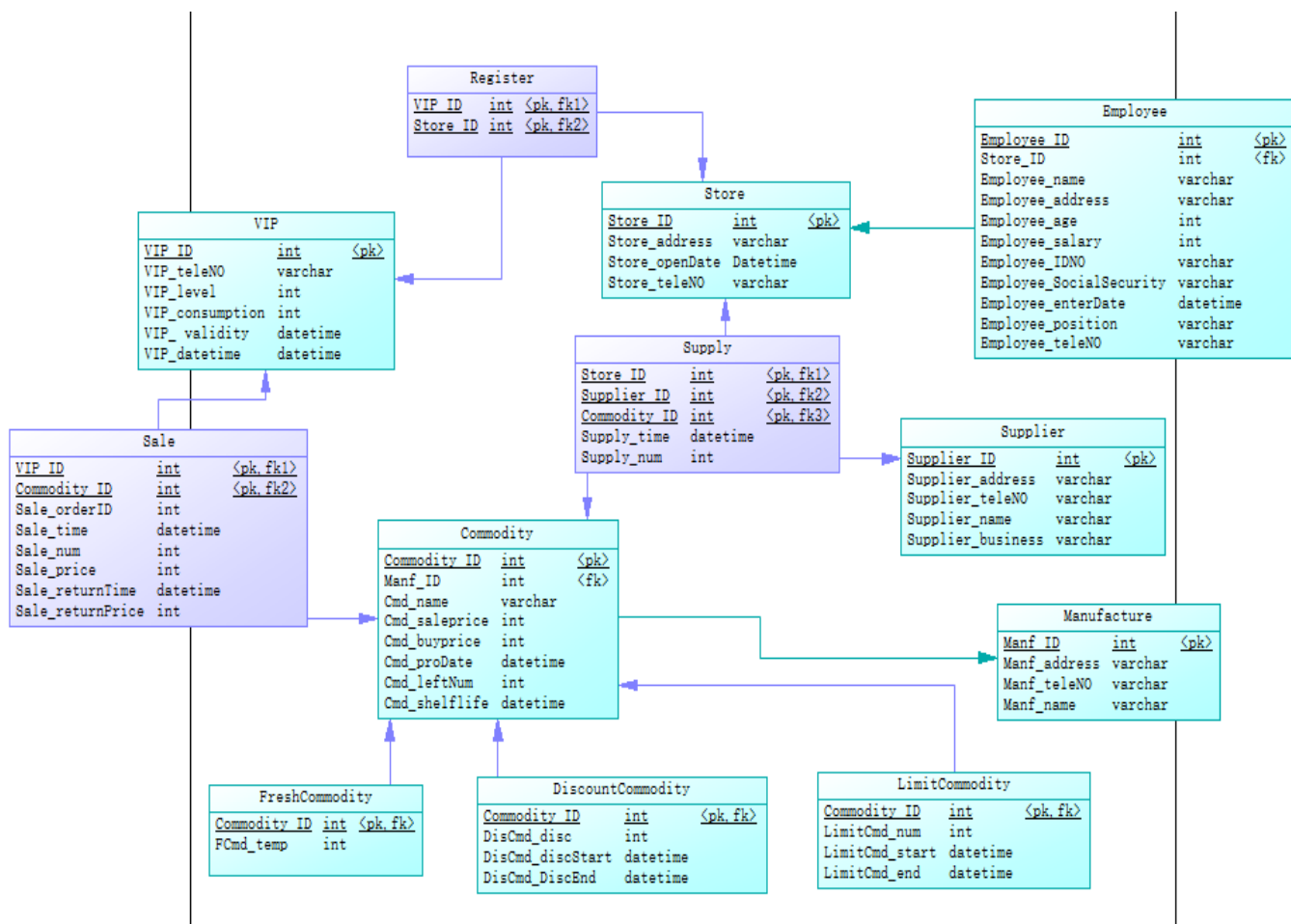
三、使用 Powerdesigner 工具实现数据库设计:

1 画出连锁超市概念模型 ER 图截图如下:



图表 15 连锁便利店数据库 ER 图示

2 转化的关系模式截图如下图所示：



图表 16 便利店数据库关系模式图示

3 使用 powerdesigner 生成的 SQL 语句，由于语句数量较多，不便于在作业中贴出，故附在附件“PowerDesignerCreate_SQL.sql”文件中，烦请老师查看。

见附件 [PowerDesignerCreate_SQL.sql](#)

四、分析比较上述两种方法：

1 两种关系模式设计存在的差异以及对于后期实现带来的不同影响：

答：这两种关系模式的设计基本一致，不存在特别大的差异，不过第一种 ER 概念图使用方框表示实体，棱形表示联系，椭圆表示属性，而 PowerDesigner 中 ER 概念图使用方框表示实体及其属性，使用连线以及椭圆形表示包含属性的联系，使用带标号连线表示无属性联系。不过这点微小的差异对于后期的实现没有什么影响。

2 Powerdesigner 工具生成的 SQL 语句特点以及附加语句出现的原因和作用：

答：①PowerDersigner 生成的 SQL 语句注释十分清楚，每一个关系模式定义以及重要步骤均有注释说明，这样做便于代码的维护；

②在定义模式之前会进行检查，检查数据库中是否已经存在和目前需要定义的关系模式相冲突的关系模式，若存在就会删除原来的关系模式，这样做的目的在于避免在执行过程中因冲突报错而中断关系模式的创建；

③自动生成的 SQL 语句在 Create Table xxx() 语句中只定义各种属性和主键，外键和

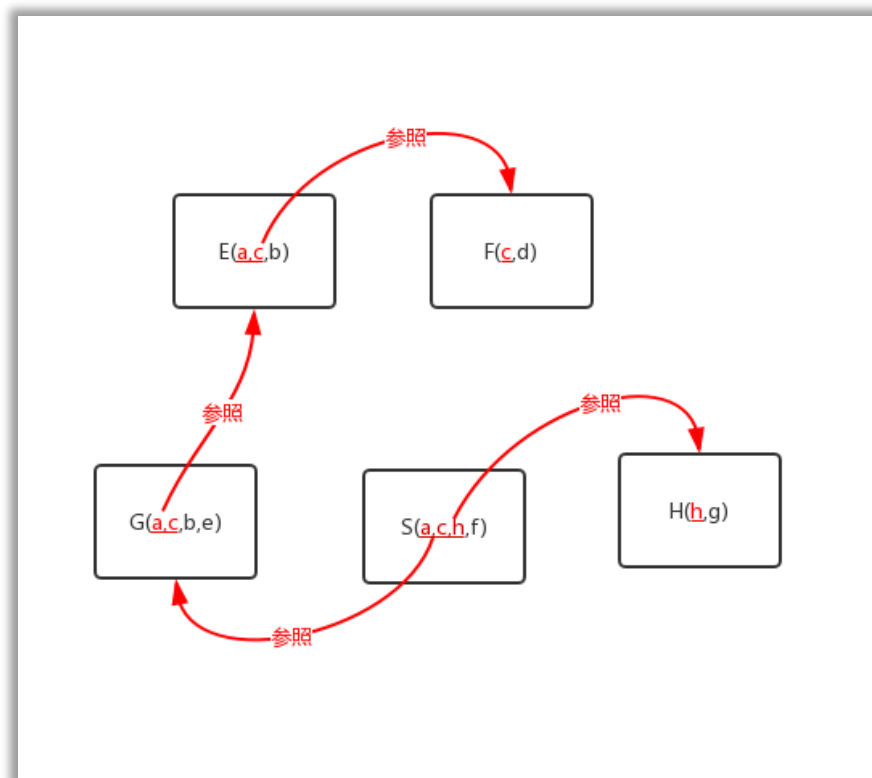
参照都是之后在声明关系模式语句外添加的，这样有利于区分不同重要性的代码，便于修改和维护；

④Powerdesigner 生成的 SQL 语句会自动给定义的主键和外键设置具有具体语义的名称，其作用在于之后有利于进行其他相关的设置，以及检索。

第二题：转化图中的概念 ER 图为关系模式图，并使用 SQL 语句实现：

- 1 关系模式转换：关系模式(子类的设计使用 OO 的方式，红色字体带下划线表示主键，连线表示外键参照)：

E(<u>a,c</u> ,b)	F(<u>c</u> ,d)
G(<u>a,c</u> ,b,e)	
S(<u>a,c,h</u> ,f)	H(<u>h</u> ,g)



图表 17 关系模式图示

- 2 关系模式 SQL 语句实现(代码文件为附件“SQL_2.txt”)：

```
1 CREATE TABLE F(  
2     c int,  
3     d varchar(20),  
4     primary key(c),  
5 );  
6
```

```
7 CREATE TABLE E(  
8     a int,  
9     c int,  
10    b varchar(20),  
11    primary key(a, c),  
12    foreign key(c) references F(c)  
13 );  
14  
15 CREATE TABLE G(  
16     a int,  
17     c int,  
18     b varchar(20),  
19     e varchar(20),  
20     primary key(a, c),  
21     foreign key(a, c) references E(a, c),  
22 );  
23  
24 CREATE TABLE H(  
25     H int,  
26     G varchar(20),  
27     primary key(h),  
28 );  
29  
30 CREATE TABLE S(  
31     a int,  
32     c int,  
33     h int,  
34     f varchar(20),  
35     primary key(a, c, h),  
36     foreign key(a, c) references G(a, c),  
37     foreign key(h) references H(h)  
38 );
```